

### **AMENDMENTS TO THE CLAIMS**

Please amend the claims as follows:

#### **Listing of Claims:**

Claim 1 (Currently Amended): A method of assigning a plurality of threads to a plurality of processors, said plurality of processors are connected to a shared memory, each of the threads being a unit of execution of a real-time operation, the method comprising:

selecting a tightly coupled thread group from among the threads based on coupling attribute information indicative of a coupling attribute between the threads, the tightly coupled thread group including a set of tightly coupled threads running in cooperation with each other; and

reserving execution terms of the tightly coupled threads in several processors of the plurality of the processors, the reserved execution terms having the same execution start timing and the same term, the several processors being equal in number to the tightly coupled threads; ~~and~~

simultaneously executing the tightly coupled threads in reserved execution terms by the several processors;

selecting a loosely coupled thread group from among the threads based on the coupling attribute information, the loosely coupled thread group including a set of loosely coupled threads communicating through a buffer on the shared memory, and

performing the scheduling operation including dispatching the loosely coupled threads to one or more of the processors in accordance with a relationship in input and output between the loosely coupled threads.

Claim 2 (Original): The method according to claim 1, wherein each of said plurality of processors includes a local memory, and the method further comprises mapping the local

memory of one of the several of the processors, which executes one of the tightly coupled threads, in part of an effective address space of other one of the tightly coupled threads executed by other one of the several of the processors.

Claim 3 (Canceled).

Claim 4 (Original): The method according to claim 1, wherein each of the tightly coupled threads has context information indicating contents of a register and a local memory of one of the several of the processors.

Claim 5 (Cancelled).

Claim 6 (Original): The method according to claim 1, wherein the scheduling operation is performed by an operating system executed by one of said plurality of processors.

Claims 7-8 (Canceled).

Claim 9 (Currently Amended): A real-time processing system that executes a plurality of threads, each of the threads being a unit of execution of a real-time operation, comprising:

a plurality of processors connected to a shared memory;

a selecting section configured to select a tightly coupled thread group from among the threads based on coupling attribute information indicative of a coupling attribute between the

threads, the tightly coupled thread group including a set of tightly coupled threads running in cooperation with each other;~~and~~

a reserving section configured to reserve execution terms of the tightly coupled threads in several processors of the plurality of the processors, the reserved execution terms having the same execution start timing and the same term, the several processors being equal in number to the tightly coupled threads;~~and~~

an executing section configured to simultaneously execute the tightly coupled threads in reserved execution terms by the several processors;

means for selecting a loosely coupled thread group from among the threads based on the coupling attribute information, the loosely coupled thread group including a set of loosely coupled threads communicating through a buffer on the shared memory; and

means for dispatching the loosely coupled threads to one or more of the processors in accordance with a relationship in input and output between the loosely coupled threads.

Claim 10 (Original): The real-time processing system according to claim 9, wherein each of said plurality of processors includes a local memory, and the system further comprises means for mapping the local memory of one of the several of the processors, which execute one of the tightly coupled threads, in part of an effective address space of other one of the tightly coupled threads executed by other one of the several of the processors.

Claim 11-14 (Canceled).

Claim 15 (Currently Amended): A computer readable storage medium including computer executable instructions, wherein the instructions, when executed by a plurality of

processors, perform a scheduling operation to assign a plurality of threads to the processors, said plurality of processors are connected to a shared memory, wherein each of the threads being a unit of execution of a real-time operation, the method comprising:

causing the computer to select a tightly coupled thread group from among the threads based on coupling attribute information indicative of a coupling attribute between the threads, the tightly coupled thread group including a set of tightly coupled threads running in cooperation with each other;~~and~~

reserving execution terms of the tightly coupled threads in several processors of the plurality of the processors, the reserved execution terms having the same execution start timing and the same term, the several processors being equal in number to the tightly coupled threads;~~and~~

simultaneously executing the tightly coupled threads in reserved execution terms by the several processors;

selecting a loosely coupled thread group from among the threads based on the coupling attribute information, the loosely coupled thread group including a set of loosely coupled threads communicating through a buffer on the shared memory, and

performing the scheduling operation including dispatching the loosely coupled threads to one or more of the processors in accordance with a relationship in input and output between the loosely coupled threads.

Claim 16 (Original): The program according to claim 15, wherein each of said plurality of processors includes a local memory, and the program further comprises causing the computer to map the local memory of one of the several of processors that execute one of the tightly coupled threads, in part of an effective address space of other one of the tightly coupled threads executed by other one of the several of the processors.